

Συμβολοσειρές - Μέρος 2ο

Μιχάλης Δρακόπουλος

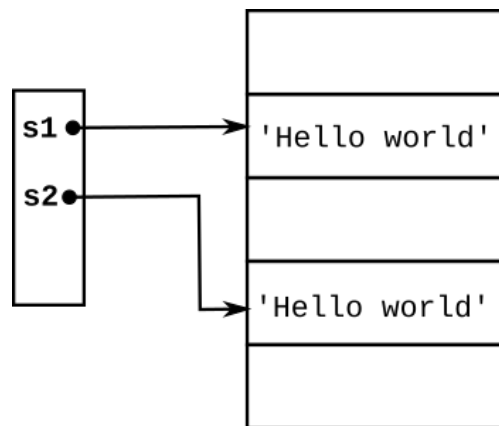
Μάρτιος 2021

Αντικείμενα

Ισότητα vs Ταυτότητα αντικειμένων

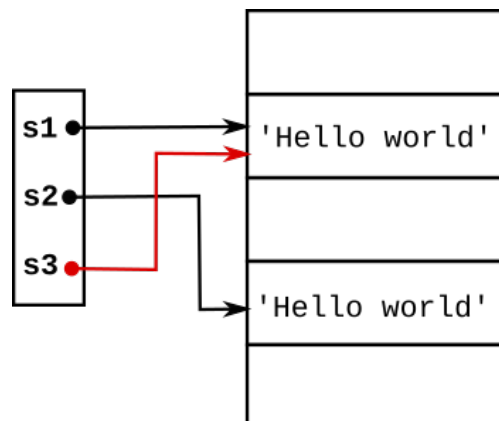
```
[1]: s1 = 'Hello world'
      s2 = 'Hello world'
```

Δημιουργούνται 2 **διαφορετικά** αντικείμενα-συμβολοσειρές



```
[2]: s3 = s1
```

Δημιουργείται μια **νέα αναφορά** στο υπάρχον αντικείμενο που είναι συνδεδεμένο με τη μεταβλητή s1. Οι μεταβλητές s1 και s3 **αναφέρονται** στο **ίδιο** αντικείμενο.



- Ο τελεστής `==` ελέγχει για **ισότητα**: αν 2 αντικείμενα είναι ίσα μεταξύ τους (για συμβολοσειρές, αν έχουν ακριβώς τα ίδια στοιχεία)

```
[3]: s1 == s2, s2 == s3, s1 == s3
```

```
[3]: (True, True, True)
```

- Ο τελεστής `is` ελέγχει για **ταυτότητα**: αν 2 αντικείμενα ταυτίζονται!

```
[4]: s1 is s2, s2 is s3, s1 is s3
```

```
[4]: (False, False, True)
```

Μέθοδοι

Στον αντικειμενοστραφή προγραμματισμό, μια **μέθοδος** είναι μια εξειδικευμένη συνάρτηση που σχετίζεται με κάποιο αντικείμενο και αντιστοιχεί στις ενέργειες που μπορούμε να κάνουμε **με** το αντικείμενο αυτό ή **πάνω** στο αντικείμενο αυτό.

Οι μέθοδοι έχουν ειδικό *τρόπο κλήσης*: Η μέθοδος `method` καλείται για το αντικείμενο `obj` με ορίσματα `args` με τον τελεστή επιλογής `.` (τελεία).

```
obj.method(args)
```

Συμβολοσειρές (συνέχεια)

Δυο ακόμα τελεστές

- `in` και `not in` : Ελέγχουν αν μια συμβολοσειρά περιέχεται σε μια άλλη

```
[5]: 'onty' in 'Monty Python'
```

```
[5]: True
```

```
[6]: 'pyt' not in 'Monty Python'
```

```
[6]: True
```

```
[7]: 'Pyt' not in 'Monty Python'
```

```
[7]: False
```

Μέθοδοι για συμβολοσειρές

Ελεγχος περιεχομένου

Επιστρέφουν `True` ή `False`.

- `isalnum()`, `isalpha()`, `isdigit()`, `islower()`, `isupper()`, `isspace()` : Ελέγχουν το είδος των στοιχείων μιας συμβολοσειράς.

```
[8]: robot1 = 'R2d2'
     robot2 = 'c3po'
```

```
[9]: robot1.isalnum(), robot1.isalpha()
```

```
[9]: (True, False)
```

```
[10]: '42'.isdigit(), robot2.islower(), robot2.isupper()
```

```
[10]: (True, True, False)
```

```
[11]: '\n \t\n '.isspace()
```

```
[11]: True
```

- startswith(substr) / endswith(substr) : Ελέγχουν αν η συμβολοσειρά αρχίζει / τελειώνει στην συμβολοσειρά substr

```
[12]: filename = input('Give a filename: ')
     if filename.endswith('.py'):
         print(filename, 'is a Python file')
     elif filename.endswith('.docx'):
         print(filename, 'is a Word document')
```

Give a filename: spam.py

spam.py is a Python file

Τροποποιητικές μέθοδοι

Επιστρέφουν **αντίγραφο** της συμβολοσειράς με κατάλληλες αλλαγές. - lower() / upper() : μετατροπή σε πεζά / κεφαλαία

```
[13]: movie = 'The Life of Brian'
     new = movie.upper()
     print(new, movie.lower(), movie, sep='\n')
```

THE LIFE OF BRIAN

the life of brian

The Life of Brian

Παράδειγμα Ελεγχος ισότητας συμβολοσειρών χωρίς διάκριση πεζών/κεφαλαίων

```
[14]: while True:
     s = input('Type Yes to continue, Quit with anything else')
     if s.upper() == 'YES':
         print('Continuing')
     else:
         print('Bye!')
         break
```

Type Yes to continue, Quit with anything else YES

Continuing

Type Yes to continue, Quit with anything else yes

Continuing

Type Yes to continue, Quit with anything else Yes

Continuing

Type Yes to continue, Quit with anything else yES

Continuing

Type Yes to continue, Quit with anything else yEs

Continuing

Type Yes to continue, Quit with anything else qqQ

Bye!

- `rstrip()`, `rstrip()`, `strip()` : Αφαιρεί λευκούς χαρακτήρες από αρχή, τέλος, αρχή και τέλος

```
[15]: lesson = ' \t\n\t\t\t Computer Science 2 \n\t '
```

```
print('1. Of all lessons', lesson.rstrip(), 'is the most boring')
```

```
print('2. Of all lessons', lesson.rstrip(), 'is the most interesting')
```

```
print('3. Of all lessons', lesson.strip(), 'is the most useful')
```

1. Of all lessons Computer Science 2

is the most boring

2. Of all lessons

Computer Science 2 is the most interesting

3. Of all lessons Computer Science 2 is the most useful

- `rstrip(c)`, `rstrip(c)`, `strip(c)` : Αφαιρεί τους χαρακτήρες της συμβολοσειράς `c` από αρχή, τέλος, αρχή και τέλος

```
[16]: m = 'speed v = 10m/s'
```

```
print(m.rstrip('pse d'))
```

v = 10m/s

```
[17]: print(m.rstrip('ms/'))
```

speed v = 10

```
[18]: print(m.strip('pse d/'))
```

v = 10m

Πλήθος εμφανίσεων, αναζήτηση, αντικατάσταση

- `count(substr)` : πλήθος εμφανίσεων της `substr` στη συμβολοσειρά

```
[19]: fruit = 'Banana'
      fruit.count('a'), fruit.count('an'), fruit.count('b')
```

```
[19]: (3, 2, 0)
```

- `find(substr)` : Επιστρέφει την θέση πρώτης εμφάνισης της `substr` στη συμβολοσειρά, ή -1

```
[20]: fruit.find('na'), fruit.find('ban')
```

```
[20]: (2, -1)
```

- `replace(old, new)` : αντικατάσταση όλων των `old` με `new` στη συμβολοσειρά

```
[21]: fruit.replace('an', 'aNN')
```

```
[21]: 'BaNNaNNa'
```

```
[22]: fruit
```

```
[22]: 'Banana'
```

Παράδειγμα: Έλεγχος νέου password

Σε μια εφαρμογή, ένα νέο password είναι αποδεκτό αν: - έχει περισσότερους από 10 χαρακτήρες - περιέχει μόνο γράμματα και αριθμούς - περιέχει τουλάχιστον ένα από τα εξής: κεφαλαίο γράμμα, πεζό γράμμα, ψηφίο - ξεκινάει από γράμμα

```
[23]: passwd = input('Enter new password: ')
      is_valid = False
      if len(passwd) > 10 and passwd.isalnum() and passwd[0].isalpha():
          has_upper = has_lower = has_digit = False
          for ch in passwd:
              if ch.isupper():
                  has_upper = True
              elif ch.islower():
                  has_lower = True
              elif ch.isdigit():
                  has_digit = True
          if has_upper and has_lower and has_digit:
              is_valid = True
      if is_valid:
          print('Password OK!')
      else:
          print('Password rejected!')
```

```
Enter new password: my1stNewPasswOrd
```

```
Password OK!
```

Μορφοποίηση συμβολοσειρών: f-strings

```
[24]: first = 'Isaac'
      last = 'Newton'
      message = f"{last}'s first name was {first}"           # Newton's first
      ↪ name was Isaac
      message
```

```
[24]: "Newton's first name was Isaac"
```

```
[25]: print(message)
```

Newton's first name was Isaac

```
[26]: fact = F'{first[0]}. {last.upper()} likes apples'
      print(fact)
```

I. NEWTON likes apples

```
[27]: import math
      R = 10
      line = f'The perimeter of circle with R = {R} is {2*math.pi*R}'
      print(line)
```

The perimeter of circle with R = 10 is 62.83185307179586

```
[28]: R = 10/3
      line = f'The perimeter of circle with R = {R:.3f} is {2*math.pi*R:12.1f}'
      print(line)
```

The perimeter of circle with R = 3.333 is 20.9